A Modular Framework for Iterative Combinatorial Auctions

A Modular Framework for Iterative Combinatorial Auctions

SÉBASTIEN LAHAIE
Yahoo Research
and
DAVID C. PARKES
Harvard University

We describe a modular elicitation framework for iterative combinatorial auctions. The framework includes proxy agents, each of which can adopt an individualized bidding language to represent partial value information of a bidder. The framework leverages algorithms from query learning to elicit value information via bids as the auction progresses. The approach reduces the multi-agent elicitation problem to isolated, single-agent learning problems, with competitive equilibrium prices used to facilitate auction clearing even without complete learning.

Categories and Subject Descriptors: J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms: Algorithms, Economics

Additional Key Words and Phrases: combinatorial auction, preference elicitation, query learning

In a combinatorial auction, bidders place bids on packages rather than just individual items in order to account for complements or substitutes. The study of combinatorial auctions has blossomed in recent years, driven by important applications in both the public and private sectors, ranging from spectrum allocation to supply-chain management. Combinatorial auctions can be broadly categorized into *single-shot* and *iterative* auctions. The distinction is important because iterative auctions, unlike single-shot auctions, allow for coordinated preference elicitation coupled with price discovery.

In a single-shot auction bidders report their preferences in some *bidding language*. The auctioneer then computes an allocation and payments and the auction ends. The winner-determination problem is typically solved in practice through mixed-integer programming (MIP) techniques, and because of this bidding languages are often designed so that they have straightforward conversions to MIP formulations. This is true of bidding languages proposed in the AI literature: languages formulated in terms of "atomic bids" (bids on single bundles) such as XOR, OR, and OR* [Nisan 2000]; languages that use logical connectives to describe complements

Authors' addressses: lahaies@yahoo-inc.com, parkes@eecs.harvard.edu

and substitutes [Boutilier and Hoos 2001; Cavallo et al. 2005]; and practical languages that allow side constraints and non-price attributes [Sandholm and Suri 2006]. The flexibility of MIP formulations leads to a wealth of plausible bidding languages.

In an iterative auction bidders communicate claims about value information in the form of bids over several rounds, typically in response to price feedback. Iterative combinatorial auctions are commonly adopted in practical applications, for example in the context of sourcing [Sandholm 2007], because of the advantages that they provide in guiding bidders towards packages of items on which they are especially competitive. But in contrast to single-shot designs, and despite the practical importance of iterative designs, the existing literature related to iterative auctions has considered only a few bidding languages. The auction designs that provide provable guarantees on efficiency and convergence are restricted to the XOR language, in the sense that the partial value information accumulated across rounds, in the form of bids, is represented in XOR [Ausubel and Milgrom 2002; de Vries et al. 2007; Parkes 1999]. But in some settings this may not be practical; for instance, XOR representations are exponential in size when valuations are additive over small, disjoint sets of items. The prevalence of XOR in iterative combinatorial auctions is an artifact of the way they are often designed: an iterative auction can be interpreted as a dual method on a linear program for the allocation problem, and a linear programming formulation leads naturally to XOR representation of bids and prices [de Vries et al. 2007; Parkes and Ungar 2000].

In recent work, we have developed a modular framework for iterative combinatorial auctions that aims to overcome this restriction [Lahaie 2007; Lahaie et al. 2005; Lahaie and Parkes 2004]. In our work, we view an iterative auction as a way of performing coordinated *learning* of the individual bidders' valuations for the purpose of identifying an efficient allocation. We are interested in learning the valuations only to the extent needed to identify an efficient allocation. In general, it will be possible to terminate before the bidder valuations are completely learned. The general framework is given in Figure 1. It consists of four components: the bidders with their valuation functions $v = (v_1, \ldots, v_n)$; proxies associated to each of the bidders, which hold partial valuation information $\tilde{v}$ in an appropriate structured representation; a winner-determination engine that takes the partial valuations and computes a tentative allocation S; and a pricing engine that computes prices p to tentatively clear the market.

The proxies issue queries to their agents to form an estimate of the agent valuations. At well-defined breakpoints, the proxies halt their elicitation and forward their estimates to the winner-determination engine, which computes an efficient allocation with respect to these reports. The allocation and reports are then forwarded to the pricing engine, which computes market-clearing prices. Finally, the allocation and prices are communicated to proxies and on to agents as necessary. If the market clears—each agent's allocated bundle maximizes its utility at the quoted prices—then the auction terminates; otherwise the elicitation process repeats.

The elicitation component of our framework is completely modular. As our figure suggests, the choice of representation language can differ across proxies. Proxies interact with their respective agents through a standard interface of *value* and *de-*

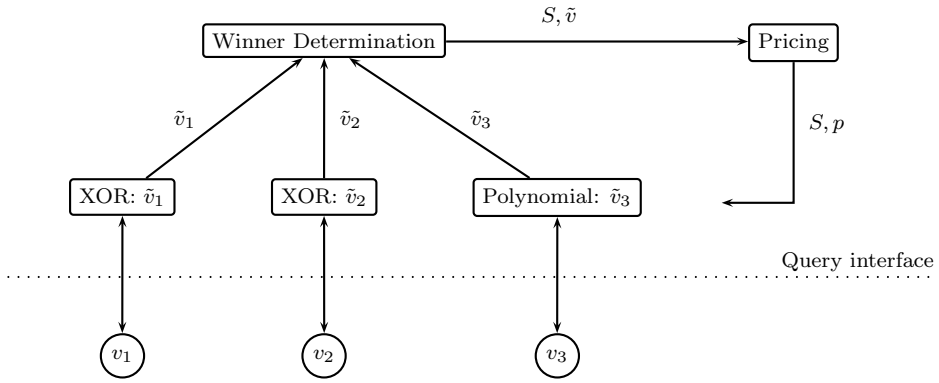

Fig. 1. Sketch of the framework.

mand queries. On a value query, the agent is a given a bundle and replies with a claim of its value for the bundle. On a demand query, the agent is given prices and replies with a claim of the bundle that maximizes its utility. Both are natural in economic contexts, and our choice also allows us to draw on algorithms for active learning in *computational learning theory* [Kearns and Vazirani 1994]. This literature typically considers *value* and *equivalence* queries [Angluin 1987]. On an equivalence query, the agent is given the estimate $\tilde{v}_i$; the agent either confirms that $\tilde{v}_i = v_i$, or provides some bundle as a counterexample to this fact. The key insight is that, in an auction problem, equivalence queries can be replaced with demand queries using the allocation and prices computed at each stage. These are sufficient to simulate equivalence queries until the efficient allocation is known. The demand queries in our approach are derived with the larger multi-agent process in mind, because prices are computed with global knowledge of all estimated valuations $\tilde{v}_i$. Thus the result is truly a multi-agent elicitation process rather than just isolated, single-agent learning processes with price-based queries. To date there are polynomial-query learning algorithms for XOR, OR, and Polynomials [Lahaie et al. 2005; Lahaie and Parkes 2004; Schapire and Sellie 1996]. If our choice of queries seems restrictive, we also note that demand queries are quite powerful and can simulate several other types of queries [Blumrosen and Nisan 2005]. Also, our demand queries may quote personalized bundle prices, as opposed to item-based prices, which limit the kinds of languages that can be efficiently learned [Blum et al. 2004].

The framework also provides flexibility in the choice of allocation algorithm, as long as it properly interfaces with the proxies. The winner-determination problem is typically formulated and solved as a MIP, and this is the approach we adopt in our framework. We therefore require that the problem of evaluating a value query on each proxy's representation $\tilde{v}_i$ have a succinct formulation as a *maximization* MIP. This allows the representations to be integrated into a MIP formulation of the winner-determination problem.

The current options for the pricing engine are more limited. The simplest approach is to use the estimates $\tilde{v}$ themselves as tentative clearing prices, or $\tilde{v}$ discounted by "core payoffs" for the agents [Ausubel and Milgrom 2002]. One could also use item prices in early rounds to drive forward elicitation. It is important to ensure that bidders have a practical way to compute responses to demand queries when faced with prices. For this, we define the notion of a *pricing language*, not evident to date in the literature. In analogy to our requirement on bidding languages, instances of a pricing language should allow for succinct *minimization* MIP formulations of value queries. This allows agents that represent their valuations internally in some bidding language (not necessarily the same as the proxy language) to formulate and solve a demand query as a MIP. XOR and Polynomials are suitable pricing languages, but to date this does not seem to be the case for OR. Of course, if agents already have bidding language representations of their valuations, one might wonder why we need elicitation at all. Note though that winner-determination algorithms tailored to the agents' representations may not be available. With our modular framework, we can recover agent valuations in a representation suitable for the winner-determination engine, which introduces modularity in the components.

Modularity in the proxies' bidding languages can bring about significant benefits in performance. To illustrate this, we performed experiments where each agent's valuation had a succinct OR representation, generated from legacy distributions in the CATS test suite [Leyton-Brown et al. 2000]. We tested two instantiations of our framework, one with XOR representations at the proxies, the other with Polynomial representations. Because we used $\tilde{v}$ as prices in each round, the proxy representations had to be drawn from a language suitable for both bidding and pricing, which excluded OR. We tuned a parameter that, roughly, determines the degree to which items are complements or substitutes. For settings where items are mostly complements, the Polynomial framework converges in under a minute, whereas the XOR framework does not converge after 24 hours. This reflects the fact that the agents' underlying valuations have succinct Polynomial representations, but the smallest XOR representations are exponential in size, and thus even the partial representations $\tilde{v}$ become unwieldy with XOR. The reverse situation holds for settings where items are substitutes: XOR can perform dramatically better. In this sense the two languages complement each other, and this exemplifies the modularity of our approach: with modest insight into the complement or substitute properties of the agents' valuations, an auctioneer can choose an appropriate language at each proxy.

We view this work as providing a first step towards more modular iterative combinatorial auctions that can provide the flexibility in bidding languages that has proved useful in single-shot designs. There is still much work to be done. The current catalog of languages for which learning algorithms are available is limited, and the pricing methodology is still rudimentary. The advantage of the framework is that the multi-agent elicitation problem can be reduced to isolated, single-agent learning problems using familiar queries. We hope that this divide-and-conquer approach will help drive forward progress on the elicitation aspects of iterative combinatorial auctions.

REFERENCES

Angluin, D. 1987. Queries and concept learning. *Machine Learning 2*, 319–342.

Ausubel, L. M. and Milgrom, P. R. 2002. Ascending auctions with package bidding. *Frontiers of Theoretical Economics 1*, 1–42.

Blum, A., Jackson, J., Sandholm, T., and Zinkevich, M. 2004. Preference elicitation and query learning. *Journal of Machine Learning Research 5*, 649–667.

Blumrosen, L. and Nisan, N. 2005. On the computational power of iterative auctions. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*. ACM Press, 29–43.

Boutilier, C. and Hoos, H. H. 2001. Bidding languages for combinatorial auctions. In *Proceedgins of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. 1211–1217.

Cavallo, R., Parkes, D. C., Juda, A., Kirsch, A., Kulesza, A., Lahaie, S., Lubin, B., Michael, L., and Shneidman, J. 2005. TBBL: A tree-based bidding language for iterative combinatorial exchanges. In *IJCAI Workshop on Preference Handling*.

de Vries, S., Schummer, J., and Vohra, R. V. 2007. On ascending Vickrey auctions for heterogeneous objects. *Journal of Economic Theory 132,* 1, 95–118.

Kearns, M. J. and Vazirani, U. V. 1994. *An Introduction to Computational Learning Theory*. MIT Press.

Lahaie, S. 2007. A modular framework for multi-agent preference elicitation. Ph.D. thesis, Harvard University.

Lahaie, S., Constantin, F., and Parkes, D. C. 2005. More on the power of demand queries in combinatorial auctions: Learning atomic languages and handling incentives. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*. Edinburgh, Scotland.

Lahaie, S. and Parkes, D. C. 2004. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*. ACM Press, 180–188.

Leyton-Brown, K., Pearson, M., and Shoham, Y. 2000. Towards a universal test suite for combinatorial auction algorithms. In *Proceedings of the second ACM Conference on Electronic Commerce (EC)*. Minneapolis, MN, 66–76.

Nisan, N. 2000. Bidding and allocation in combinatorial auctions. In *second ACM Conference on Electronic Commerce (EC)*. Minneapolis, MN, 1–12.

Parkes, D. C. 1999. *i*Bundle: An efficient ascending price bundle auction. In *Proceedings of the first ACM Conference on Electronic Commerce (EC)*. Denver, CO, 148–157.

Parkes, D. C. and Ungar, L. H. 2000. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*. 74–81.

Sandholm, T. 2007. Expressive commerce and its application to sourcing: How we conducted $35 billion of generalized combinatorial auctions. *AI Magazine 28,* 3 (Fall), 45–58.

Sandholm, T. and Suri, S. 2006. Side constraints and non-price attributes in markets. *Games and Economic Behavior 55*, 321–330.

Schapire, R. and Sellie, L. 1996. Learning sparse multivariate polynomials over a field with queries and counterexamples. *Journal of Computer and Systems Sciences 52,* 2, 201–213.